CAMBRIDGE UNIVERSITY
ENGINEERING DEPARTMENT

SEMI-TIED COVARIANCE MATRICES FOR HIDDEN MARKOV MODELS

M.J.F. Gales

February 1998

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

E-mail: mjfg@eng.cam.ac.uk
http://www-svr.eng.cam.ac.uk/~mjfg

Abstract

There is normally a simple choice made in the form of the covariance matrix to be used with continuous-density HMMs. Either a diagonal covariance matrix is used, with the underlying assumption that elements of the feature vector are independent, or a full or block-diagonal matrix is used, where all or some of the correlations are explicitly modelled. Unfortunately when using full or block-diagonal covariance matrices there tends to be a dramatic increase in the number of parameters per Gaussian component, limiting the number of components which may be robustly estimated. This paper introduces a new form of covariance matrix which allows a few "full" covariance matrices to be shared over many distributions, whilst each distribution maintains its own "diagonal" covariance matrix. In contrast to other schemes which have hypothesised a similar form, this technique fits within the standard maximum-likelihood criterion used for training HMMs. The new form of covariance matrix is evaluated on a large-vocabulary speech-recognition task. In initial experiments the performance of the standard system was achieved using approximately half the number of parameters. Moreover, a 10% reduction in word error rate compared to a standard system can be achieved with less than a 1% increase in the number of parameters and little increase in recognition time.

1 Introduction

There is normally a simple choice made in the form of the covariance matrix to be used with continuous-density hidden Markov models (HMMs) [19]. Either a diagonal covariance matrix is used, with the underlying assumption that each element of the feature vector is independent, or a full or block-diagonal matrix is used, where all or some of the correlations are explicitly modelled. Unfortunately when using full or block-diagonal covariance matrices there tends to be a dramatic increase in the number of parameters per Gaussian component, limiting the number of components which may be robustly estimated. To overcome this problem multiple diagonal-covariance Gaussian distributions may be used [16, 13]. In addition to being able to model non-Gaussian distributions they can model correlations. However, it is preferable to decorrelate the feature vector as far as possible, as otherwise components must be used to model correlations rather than the possible non-Gaussian nature of the density function associated with a particular state.

There have been many attempts to overcome the problem of compactly modelling data where the elements of the feature vector are correlated with one another. They may be split into two classes, feature-space and model-space schemes. In feature-space schemes, the front-end processing is modified to try and ensure that all elements of the feature vector are independent. The use of the discrete cosine transform in speech recognition is common for this reason [3]. Other schemes include linear discriminant analysis (LDA) and the Karhunen-Loève transform [5]. However, it is hard to find a single transform which decorrelates all elements of the feature vector for all states. Model-based schemes are a more flexible approach, which allow many decorrelating transforms to be used. A different transform is selected depending on which component the observation was hypothesised to be generated from. In the limit a transform may be used for each component, which is equivalent to a full covariance matrix system. This paper introduces a new model-based scheme, *semi-tied* covariance matrices. The scheme which is most closely related to the one described in this paper is the state-specific rotation [17], which normally uses a separate transform for each state, but may be applied at any level of clustering.

The model-space transform introduced in this paper is a natural extension of the state-specific rotation scheme. Instead of estimating the transform independently of the specific components associated with it, the transform is estimated in a maximum-likelihood (ML) fashion given the current model parameters. This optimisation is performed using a simple iterative scheme, which is guaranteed to increase the likelihood of the training data. Recently an extension to LDA based on ML has been proposed [14], heteroscedastic LDA (HLDA). Although addressing a different problem, that of dimensionality reduction, optimising the HLDA transform requires solving similar equations to the ones described here. In contrast to the scheme presented here numerical techniques or steepest descent are used in estimating the transform. With a simple modification the optimisation scheme described here may be used to obtain the linear discriminant transform for the diagonal covariance matrix case. Furthermore, the two approaches can be combined so that each transformation selects a particular feature sub-space dependent on the hypothesised component, rather than just a global linear transformation of the feature space. The optimisation technique described may also be used for solving the problem of adapting covariance matrices in a speaker, or environmental, adaptation task.

The next section describes the state-specific rotation and related schemes. Semi-tied covariance matrices are then introduced and re-estimation formulae, which are guaranteed to increase the likelihood of the training data, are detailed. In section 4, the semi-tied covariance optimisation problem is related to the problem of optimising the HLDA transform and the ML variance adaptation problem. Various implementation issues, such as the memory requirements, how the component to transform clustering may be performed and numerical accuracy issues are discussed. The use of standard linear model-space adaptation schemes in conjunction with the semi-tied covariance matrices is then described. The new technique is evaluated on a large-vocabulary speech recognition task.

2 State-Specific Rotations

In HMM-based systems there is a basic choice in the form of covariance matrix to be used. It may either be diagonal, block-diagonal, or full. The full covariance matrix case has the advantage over the diagonal case in that it models inter feature-vector element correlation. However this is at the cost of a greatly increased number of parameters, $n(n+3)/2$ compared to $2n$ per component, including the mean vector and the covariance matrix, where n is the dimensionality. Due to this massive increase in the number of parameters, diagonal covariance matrices are commonly used in large-vocabulary speech recognition. The data associated with each state is modelled by multiple Gaussian components. The hope is that by using multiple components any strong correlations may be implicitly modelled, in addition to the possible non-Gaussian nature of the data. However, if the correlations in the data could be explicitly modelled, then either the number of Gaussian components per state could be reduced, hence reducing the size of the model sets, or more accurate recognition could be achieved by allowing the Gaussian components to model the non-Gaussian nature of the data, rather than the correlations.

One scheme proposed for modelling the correlations in the feature vector is to use state-specific rotations [17]. Here a full covariance matrix is calculated for each state in the system. This is then decomposed into its eigenvectors and eigenvalues. All data from that state is then decorrelated using the eigenvectors calculated. Multiple diagonal covariance matrix Gaussian components are then trained. Thus, the covariance matrix associated with each state, s , is decomposed as

$$\mathbf{\Sigma}^{(s)}_{\text{full}} = \mathbf{U}^{(s)}\mathbf{\Lambda}^{(s)}\mathbf{U}^{(s)T} \tag{1}$$

where

$$\mathbf{\Sigma}^{(s)}_{\text{full}} = \frac{\sum_\tau \gamma_s(\tau)(\mathbf{o}(\tau)-\mu^{(s)})(\mathbf{o}(\tau)-\mu^{(s)})^T}{\sum_{\tau=1}^{T}\gamma_s(\tau)} \tag{2}$$

$\mathbf{U}^{(s)}$ is the matrix of eigenvectors, $\mathbf{\Lambda}^{(s)}$ is the diagonal matrix of the eigenvalues, the superscript T means matrix transpose, $\mu^{(s)}$ is the state mean and

$$\gamma_s(\tau) = p(q_s(\tau)|\mathcal{M},\mathbf{O}_T) \tag{3}$$

where $q_s(\tau)$ indicates state (or component) s at time τ and $\mathbf{O}_T$ is the complete set of training data. When training, instead of using the standard observation vector, $\mathbf{o}(\tau)$, a state specific observation vector, $\mathbf{o}^{(s)}(\tau)$, is used where

$$\mathbf{o}^{(s)}(\tau) = \mathbf{U}^{(s)T}\mathbf{o}(\tau) \tag{4}$$

Each component, m, associated with that particular state, s, is then trained using

$$\mu^{(sm)} = \frac{\sum_\tau \gamma_m(\tau)\mathbf{o}^{(s)}(\tau)}{\sum_\tau \gamma_m(\tau)} \tag{5}$$

(note that $\mu^{(sm)} = \mathbf{U}^{(s)T}\mu^{(m)}$) and

$$\mathbf{\Sigma}^{(m)}_{\text{diag}} = \text{diag}\left(\frac{\sum_\tau \gamma_m(\tau)\left(\mathbf{o}^{(s)}(\tau)-\mu^{(sm)}\right)\left(\mathbf{o}^{(s)}(\tau)-\mu^{(sm)}\right)^T}{\sum_\tau \gamma_m(\tau)}\right) \tag{6}$$

where diag(.) just extracts the leading diagonal. The covariance matrix associated with each component is

$$\mathbf{\Sigma}^{(m)} = \mathbf{U}^{(s)}\mathbf{\Sigma}^{(m)}_{\text{diag}}\mathbf{U}^{(s)T} \tag{7}$$

During recognition and training the likelihood used for component m of state s is

$$\mathcal{L}\left(\mathbf{o}(\tau);\mu^{(m)},\mathbf{\Sigma}^{(m)},\mathbf{U}^{(s)}\right) = \mathcal{N}\left(\mathbf{o}^{(s)}(\tau);\mu^{(sm)},\mathbf{\Sigma}^{(m)}_{\text{diag}}\right) \tag{8}$$

Computationally this is relatively efficient as it is only necessary to perform one rotation per state, in contrast to standard full covariance matrices which are the equivalent of one rotation per component.

Although this does partially handle the problem of modelling correlations in the feature vectors, it does not fit within the standard ML estimation framework for training HMMs. The transforms are not related to the multiple-component models being used to model the data. One simple extension is to use the average within-component covariance per state, as opposed to the global state covariance. Thus the same transform is used except that equation 2 is replaced by

$$\mathbf{\Sigma}^{(s)}_{\text{full}} = \frac{\sum_{m \in M^{(s)},\tau} \gamma_m(\tau)(\mathbf{o}(\tau) - \mu^{(m)})(\mathbf{o}(\tau) - \mu^{(m)})^T}{\sum_{m \in M^{(s)},\tau} \gamma_m(\tau)} \tag{9}$$

where $M^{(s)}$ is the set of Gaussian components used to model state s and $\mu^{(m)}$ is the current estimate of the component mean. This still does not yield a transform that is guaranteed to increase the likelihood (it uses the same sort of approximation as least-squares linear regression [11]), but does relate the transform to the current model set.

A further modification can be used to generate a transform that is guaranteed to increase the likelihood. This transform has a similar form to the variance transform described in [10]. The component-specific variance may be written as

$$\mathbf{\Sigma}^{(m)} = \mathbf{L}^{(m)}_{\text{diag}} \mathbf{\Sigma}^{(s)\prime}_{\text{full}} \mathbf{L}^{(m)T}_{\text{diag}} \tag{10}$$

where

$$\mathbf{\Sigma}^{(s)\prime}_{\text{full}} = \frac{\sum_{m \in M^{(s)}} \mathbf{L}^{(m)-1}_{\text{diag}} \left(\sum_\tau \gamma_m(\tau)(\mathbf{o}(\tau) - \mu^{(m)})(\mathbf{o}(\tau) - \mu^{(m)})^T\right) \left(\mathbf{L}^{(m)-1}_{\text{diag}}\right)^T}{\sum_{m \in M^{(s)},\tau} \gamma_m(\tau)} \tag{11}$$

and the diagonal matrix $\mathbf{L}^{(m)}_{\text{diag}}$ (in the general case this is the Choleski factorisation of the covariance matrix) is defined as

$$\mathbf{\Sigma}^{(m)}_{\text{diag}} = \mathbf{L}^{(m)}_{\text{diag}} \mathbf{L}^{(m)T}_{\text{diag}} \tag{12}$$

This allows a full-covariance element to be shared over many components. Unfortunately there is a significant increase in the computational load during recognition [10].

This paper introduces a natural extension to the state-specific rotation approach. The transforms are trained in a ML sense, whilst maintaining the low recognition-time cost of the state-specific rotation.

3 Semi-Tied Covariance Matrices

Semi-tied covariance matrices are a simple extension to the standard diagonal, block-diagonal, or full covariance matrices used with HMMs. Instead of having a distinct covariance matrix for every component in the recogniser, each covariance matrix consists of two elements, a component specific diagonal covariance element[1], $\mathbf{\Sigma}^{(m)}_{\text{diag}}$, and a *semi-tied* class dependent, non-diagonal matrix, $\mathbf{H}^{(r)}$ (referred to as the semi-tied transform). The form of the covariance matrix is then

$$\mathbf{\Sigma}^{(m)} = \mathbf{H}^{(r)} \mathbf{\Sigma}^{(m)}_{\text{diag}} \mathbf{H}^{(r)T} \tag{13}$$

[1]In the general case the component specific covariance matrix need not necessarily be diagonal, they need only be more constrained than the semi-tied transform. Though estimation formulae may be simply derived in this case (see [7] for how to estimate the semi-tied transform in the non-diagonal case), the diagonal component specific covariance case is felt to be the most practically useful and will be the one described in this paper.

$\mathbf{H}^{(r)}$ may be tied over a set of components, for example all those associated with the same state of a particular context-independent phone.

Each component, m, has the following parameters: component weight, component mean, $\mu^{(m)}$, and the diagonal element of the semi-tied covariance matrix, $\mathbf{\Sigma}^{(m)}_{\text{diag}}$. In addition it is associated with a semi-tied class, which has an associated semi-tied transform $\mathbf{H}^{(r)}$. This is used to generate the component's covariance matrix as described in equation 13. It is very complex to optimise these parameters directly so an expectation-maximisation approach is adopted [4][2]. Furthermore, rather than dealing with $\mathbf{H}^{(r)}$, it is simpler to deal with its inverse, $\mathbf{A}^{(r)}$, thus $\mathbf{A}^{(r)} = \mathbf{H}^{(r)-1}$. If ML estimates of all the parameters are made then the auxiliary function below must be optimised with respect to $\hat{\mathbf{A}}^{(r)}$, $\hat{\mu}^{(m)}$ and $\hat{\mathbf{\Sigma}}^{(m)}$

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = \tag{14}$$
$$\sum_{m\in M^{(r)},\tau} \gamma_m(\tau)\left(\log\left(\frac{|\hat{\mathbf{A}}^{(r)}|^2}{|\hat{\mathbf{\Sigma}}^{(m)}{}_{\text{diag}}|}\right) - \left(\mathbf{o}(\tau)-\hat{\mu}^{(m)}\right)^T \hat{\mathbf{A}}^{(r)T}\hat{\mathbf{\Sigma}}^{(m)-1}_{\text{diag}}\hat{\mathbf{A}}^{(r)}\left(\mathbf{o}(\tau)-\hat{\mu}^{(m)}\right)\right)$$

where $|.|$ indicates the determinant of a matrix and $M^{(r)}$ is the set of Gaussian components assigned to semi-tied class r. If all the model parameters are to be simultaneously optimised then this expression may be rewritten as[3]

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = \sum_{m\in M^{(r)},\tau} \gamma_m(\tau)\log\left(\frac{|\hat{\mathbf{A}}^{(r)}|^2}{\left|\text{diag}\left(\hat{\mathbf{A}}^{(r)}\mathbf{W}^{(m)}\hat{\mathbf{A}}^{(r)T}\right)\right|}\right) - n\beta \tag{15}$$

where

$$\mathbf{W}^{(m)} = \frac{\sum_\tau \gamma_m(\tau)\left(\mathbf{o}(\tau)-\hat{\mu}^{(m)}\right)\left(\mathbf{o}(\tau)-\hat{\mu}^{(m)}\right)^T}{\sum_\tau \gamma_m(\tau)} \tag{16}$$

$$\hat{\mu}^{(m)} = \frac{\sum_\tau \gamma_m(\tau)\mathbf{o}(\tau)}{\sum_\tau \gamma_m(\tau)} \tag{17}$$

$$\beta = \sum_{m\in M^{(r)},\tau} \gamma_m(\tau) \tag{18}$$

and $\gamma_m(\tau) = p(q_m(\tau)|\mathcal{M},\mathbf{O}_T)$ where $q_m(\tau)$ indicates component m at time τ, $\mathbf{O}_T$ is the complete set of training data and $\mathbf{o}(\tau)$ is the n dimensional observation at time τ. The ML estimate of the diagonal element of the covariance matrix is given by

$$\hat{\mathbf{\Sigma}}^{(m)}_{\text{diag}} = \text{diag}\left(\hat{\mathbf{A}}^{(r)}\mathbf{W}^{(m)}\hat{\mathbf{A}}^{(r)T}\right) \tag{19}$$

where $m \in M^{(r)}$. The re-estimation formulae for the component weights and transition probabilities are identical to the standard HMM cases [19].

Unfortunately optimising equation 15 directly is non-trivial and requires numerical optimisation techniques and a full matrix, $\mathbf{W}^{(m)}$, to be stored at each component. An alternative approach is proposed in this paper. The following scheme is used. $\mathbf{A}^{(r)}$ is initialised either with the current estimate of the semi-tied transform or an identity matrix.

[2] In the expressions to be optimised, this is indicated by making the equation a function of the previously estimated model set, $\mathcal{M}$. This model set determines the posterior Gaussian component probabilities used to estimate the new model set, $\hat{\mathcal{M}}$. It does not indicate any restriction in updating the model parameters.

[3] This uses the equality that, at the ML estimate of the mean and diagonal variance for a particular value of $\hat{\mathbf{A}}^{(r)}$, the following *minimum* value satisfies

$$\sum_{m\in M^{(r)},\tau}\gamma_m(\tau)\left(\mathbf{o}(\tau)-\hat{\mu}^{(m)}\right)^T\hat{\mathbf{A}}^{(r)T}\hat{\mathbf{\Sigma}}^{(m)-1}_{\text{diag}}\hat{\mathbf{A}}^{(r)}\left(\mathbf{o}(\tau)-\hat{\mu}^{(m)}\right) = n\sum_{m\in M^{(r)},\tau}\gamma_m(\tau)$$

1. Estimate the mean using equation 17, which is independent of the other model parameters.
2. Using the current estimate of the semi-tied transform, $\hat{\mathbf{A}}^{(r)}$, and equation 19 estimate the set of component specific diagonal variances. This set of parameters will be denoted as $\left\{\hat{\boldsymbol{\Sigma}}^{(r)}_{\text{diag}}\right\} = \left\{\hat{\boldsymbol{\Sigma}}^{(m)}_{\text{diag}}, m \in M^{(r)}\right\}$.
3. Estimate the semi-tied transform $\hat{\mathbf{A}}^{(r)}$ using the current set $\left\{\hat{\boldsymbol{\Sigma}}^{(r)}_{\text{diag}}\right\}$.
4. Goto (2) until convergence, or appropriate criterion satisfied.

At each stage the likelihood is guaranteed to increase.

Formulae to compute the ML estimates of the mean and component specific diagonal covariance matrices have been given in equations 17 and 19. However, optimising the semi-tied transform requires an iterative estimation scheme even after fixing all other model parameters. Selecting a particular row of $\hat{\mathbf{A}}^{(r)}$, $\hat{\mathbf{a}}_i^{(r)}$, (this is a 1 by n *row vector*) and rewriting equation 14 using the current set $\left\{\hat{\boldsymbol{\Sigma}}^{(r)}_{\text{diag}}\right\}$

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}; \left\{\hat{\boldsymbol{\Sigma}}^{(r)}_{\text{diag}}\right\}) = \sum_{m \in M^{(r)}, \tau} \gamma_m(\tau) \left\{ \log\left(\left(\hat{\mathbf{a}}_i^{(r)} \mathbf{c}_i^T\right)^2\right) - \log\left(|\hat{\boldsymbol{\Sigma}}^{(m)}_{\text{diag}}|\right) - \sum_j \frac{\left(\hat{\mathbf{a}}_j^{(r)} \hat{\mathbf{o}}^{(m)}(\tau)\right)^2}{\hat{\sigma}^{(m)2}_{\text{diag}j}} \right\} \tag{20}$$

where $\hat{\mathbf{o}}^{(m)}(\tau) = \mathbf{o}(\tau) - \hat{\mu}^{(m)}$, $\hat{\sigma}^{(m)2}_{\text{diag}i}$ is element i of the leading diagonal of $\hat{\boldsymbol{\Sigma}}^{(m)}_{\text{diag}}$ and $\mathbf{c}_i$ is the i^{th} row vector of the cofactors of $\hat{\mathbf{A}}^{(r)}$. This expression is then optimised for $\hat{\mathbf{A}}^{(r)}$. It can be shown that

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) \geq \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}; \left\{\hat{\boldsymbol{\Sigma}}^{(r)}_{\text{diag}}\right\}) \tag{21}$$

with equality when diagonal elements of the covariance matrix are given by equation 19, (hence the need to iterate in order to obtain an ML solution for all the parameters, since if the semi-tied transform varies, then a "better" solution for the diagonal elements is possible). In appendix A it is shown that the ML estimate for the i^{th} row of the semi-tied transform, $\hat{\mathbf{a}}_i^{(r)}$, is given by

$$\hat{\mathbf{a}}_i^{(r)} = \mathbf{c}_i \mathbf{G}^{(ri)-1} \sqrt{\left(\frac{\beta}{\mathbf{c}_i \mathbf{G}^{(ri)-1} \mathbf{c}_i^T}\right)} \tag{22}$$

where

$$\mathbf{G}^{(ri)} = \sum_{m \in M^{(r)}} \frac{1}{\hat{\sigma}^{(m)2}_{\text{diag}i}} \mathbf{W}^{(m)} \sum_\tau \gamma_m(\tau) \tag{23}$$

and $\mathbf{c}_i$ is the i^{th} row vector of cofactors of the current estimate of $\hat{\mathbf{A}}^{(r)}$. The iterative nature of this optimisation is now clear, since each row is related to the other rows by the cofactors. This is not a problem as the sufficient statistics for the optimisation are very simple, namely $\mathbf{G}^{(ri)}$ and the semi-tied class occupancy count, β.

During recognition the log-likelihood is based on [4]

$$\log\left(\mathcal{L}\left(\mathbf{o}(\tau); \mu^{(m)}, \boldsymbol{\Sigma}^{(m)}, \mathbf{A}^{(r)}\right)\right) = \log\left(\mathcal{N}\left(\mathbf{o}^{(r)}(\tau); \mathbf{A}^{(r)}\mu^{(m)}, \boldsymbol{\Sigma}^{(m)}_{\text{diag}}\right)\right) + \log\left(|\mathbf{A}^{(r)}|\right) \tag{24}$$

and

$$\mathbf{o}^{(r)}(\tau) = \mathbf{A}^{(r)} \mathbf{o}(\tau) \tag{25}$$

[4]The last term, $\log(|\mathbf{A}|)$, should strictly be written as $\frac{1}{2}\log(|\mathbf{A}|^2)$, thus allowing the determinant of $\mathbf{A}$ to go negative.

Thus by storing $\mathbf{A}^{(r)}\mu^{(m)}$ instead of $\mu^{(m)}$, the cost of calculating the likelihoods associated with semi-tied covariance matrices is that of one matrix vector multiplication per semi-tied transform class and an addition[5].

It is worth emphasising the difference between semi-tied covariance matrices and state-based rotations. The most important difference is that the semi-tied covariance matrices are trained in an ML sense on the training data given the current model set. It would only be possible to train a state-based rotation in an ML sense when all the values of $\mathbf{\Sigma}^{(m)}_{\text{diag}}$ associated with a particular transform are the same[6]. Typically this constraint is not satisfied. As the number of state-based rotations decreases, so the differences between the component specific variances associated with a particular rotation becomes larger. Hence, the difference between the state-specific rotation and ML estimated rotation also increases. Furthermore, there are no constraints on the form of the semi-tied transform in semi-tied covariance matrices. In the state-specific rotations the transforms are constrained to be orthonormal, as they are derived from the eigenvectors of the full covariance matrix associated with a state.

4 Relationship to HLDA and ML Variance Adaptation

The form of estimation routine described for semi-tied covariance matrices may be applied to other estimation problems in speech recognition and pattern matching. This section describes schemes to optimise the HLDA transform and the ML estimation of a variance transform for speaker or environmental adaptation. This section assumes that the underlying models are standard HMMs with diagonal covariance matrices, not semi-tied covariance matrices.

Heteroscedastic linear discriminant analysis (HLDA) [14] is related to semi-tied covariance matrices. HLDA is a generalisation of the standard linear discriminant analysis (LDA) scheme [5], which relaxes the assumption that all the within class covariance matrices are the same. The transform is required to reduce the dimensionality from an initial n-dimensional space to a p-dimensional space in a ML fashion, $p < n$. The objective function optimised is[7] [14]

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \sum_{m \in M, \tau} \gamma_m(\tau) \log \left(\frac{|\hat{\mathbf{A}}|^2}{\left|\text{diag}\left(\hat{\mathbf{A}}_p \mathbf{W}^{(m)} \hat{\mathbf{A}}_p^T\right)\right| \left|\text{diag}\left(\hat{\mathbf{A}}_{n-p} \mathbf{T} \hat{\mathbf{A}}_{n-p}^T\right)\right|} \right) \tag{26}$$

where

$$\mathbf{T} = \frac{1}{T} \sum_{\tau} \left(\mathbf{o}(\tau) - \mu^{(g)}\right) \left(\mathbf{o}(\tau) - \mu^{(g)}\right)^T \tag{27}$$

$\mu^{(g)}$ is the global mean of the data, $\hat{\mathbf{A}}_p$ is the first p rows of $\hat{\mathbf{A}}$ and $\hat{\mathbf{A}}_{n-p}$ are the remaining $n-p$ rows. Equation 26 is very similar to equation 14. The main difference is that for rows $j > p$ the transform acts on the global variance of the data rather than the component specific variance. By noting that the transform acts in a row by row fashion, since diagonal covariance matrices are assumed, the optimisation described in the previous section may be used. Instead of using equation 17 to fix the set of parameters $\left\{\hat{\mathbf{\Sigma}}^{(r)}\right\}$ the following expressions are used

$$\hat{\sigma}^{(m)2}_{\text{diag}j} = \begin{cases} \hat{\mathbf{a}}_j \mathbf{W}^{(m)} \hat{\mathbf{a}}_j^T & (j \le p) \\ \hat{\mathbf{a}}_j \mathbf{T} \hat{\mathbf{a}}_j^T & (j > p) \end{cases} \tag{28}$$

[5]If only one transformation class is used then $\log(|\mathbf{A}|)$ does not discriminate between the models so may be ignored. It is also possible to eliminate the requirements for the determinant term by simply scaling the elements of the diagonal covariance matrix [14].

[6]This effectively states that all component variances associated with the same state are tied.

[7]The dependence on the semi-tied class has been dropped as there is typically only one semi-tied class.

Furthermore when optimising the transform for rows ($j > p$) instead of using equation 23 use

$$\mathbf{G}^{(j)} = \sum_{m \in M} \frac{1}{\hat{\sigma}^{(m)2}_{\text{diag}j}} \mathbf{T} \sum_{\tau} \gamma_m(\tau) \tag{29}$$

in equation 22. Solutions to the case when full covariance matrices are used is also possible [7][8].

Another closely related problem is ML linear transformations of the variances for speaker and environmental adaptation [8]. Here a linear transform, typically tied over many components, is required to adapt the variances to be representative of a new speaker, or acoustic environment. When adapted in an *unconstrained* model-space fashion [8], the new variance of component m, $\hat{\mathbf{\Sigma}}^{(m)}$, may have the form

$$\hat{\mathbf{\Sigma}}^{(m)} = \mathbf{H}\mathbf{\Sigma}^{(m)}\mathbf{H}^T \tag{30}$$

where $\mathbf{H}$ is the transform to be estimated and $\mathbf{\Sigma}^{(m)}$ is the original variance. When $\mathbf{H}$ is estimated in a ML fashion, an expression identical to equation 20 must be optimised. However, now the mean estimate which is typically based on a linear transform of the original mean parameters (typically the variance is estimated after adapting the means [8]). It can therefore be optimised in the same fashion as a semi-tied transform without the need to update the component specific diagonal covariance matrices as these are typically fixed in the adaptation task[9]. At recognition time the same efficient decoding as the semi-tied models may be used.

5 Implementation Issues

5.1 Statistics required

The optimisation process described in section 3 may be run in one of two distinct modes. The choice of mode is dependent on the size of the model set being used. The memory requirements and computational load of each of the schemes is very different.

1. **Time efficient**: At each component, the occupancy, vector sum and $\mathbf{W}^{(m)}$ are stored[10]. It is then possible to optimise $\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})$ iteratively without having to examine the data again. First, $\hat{\mathbf{\Sigma}}^{(m)}_{\text{diag}}$ is estimated using the current estimate of $\hat{\mathbf{A}}^{(r)}$ and $\mathbf{W}^{(m)}$ (computational cost $\mathcal{O}(n^3)$ per component). Then, $\mathbf{G}^{(ri)}$ is found (computational cost $\mathcal{O}(n^3)$ per component) and finally the semi-tied transform estimated (computational cost $\mathcal{O}(n^4)$ per semi-tied class). The process is then repeated. In terms of computational cost this may be contrasted with using standard numerical optimisation schemes. These will usually require the calculation of the gradient given the current model parameters, an operation costing at least $\mathcal{O}(n^3)$ per component for every iteration in the optimisation. Although the iterative scheme presented here may be slightly more expensive per iteration than standard numerical optimisation techniques, in practice it converges after very few iterations (< 10). In contrast the numerical

[8] The full covariance case is only worth investigating where some reduction in the dimensionality is involved. Otherwise the semi-tied covariance matrix is subsumed in the Gaussian component's full-covariance matrix.

[9] These linear adaptation schemes have been successfully applied to adapting semi-tied models [7].

[10] This ignores the transition probability updates. Furthermore, from equation 16, $\mathbf{W}^{(m)}$ is a function of the new estimate of the mean $\hat{\mu}^{(m)}$. This may be overcome by storing the outer-product of the observation at the component level and using the standard equality

$$\mathbf{W}^{(m)} = \frac{\sum_{\tau} \gamma_m(\tau)\mathbf{o}(\tau)\mathbf{o}(\tau)^T}{\sum_{\tau} \gamma_m(\tau)} - \hat{\mu}^{(m)}\hat{\mu}^{(m)T} \tag{31}$$

For the experiments performed in this paper, where the mean was not updated when the semi-tied transform was estimated, the original mean is used when estimating $\mathbf{W}^{(m)}$.

optimisation scheme may take an order of magnitude more iterations. Furthermore, each iteration is guaranteed to increase the likelihood. Hence, there are no stability problems.

2. **Memory efficient**: For many large vocabulary speech recognition tasks it is not practical to store $\mathbf{W}^{(m)}$ for every component. To get around this problem the model parameters are estimated in multiple runs through the data. On the first run through the data, the occupancy count and vector sum for each component, and $\mathbf{G}^{(ri)}$ for each semi-tied class, are estimated using the current values of the diagonal elements of the covariance matrix. Given that in many applications there will be very few (< 100) semi-tied classes compared to the number of components (> 10000) the memory cost of storing $\mathbf{G}^{(ri)}$ per semi-tied class is very small compared to storing $\mathbf{W}^{(m)}$ per component. It is then possible to estimate the mean and semi-tied transform with the statistics stored. On the next pass through the data, the standard HMM estimation statistics (although the variance is estimated on data after applying the semi-tied transform $\hat{\mathbf{A}}^{(r)}$) are stored and the means and diagonal elements of the covariance matrix may be updated. The process is then repeated. Thus, many runs through the training data are required, but the memory requirements for the statistics tends to be no larger than training a standard model set. At each run $\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})$ is not maximised, however the likelihood is always guaranteed to increase. This form of memory efficient optimisation cannot be easily implemented with standard numerical optimisation schemes.

Variations on these schemes are possible. For example by assuming that the covariance matrices for all Gaussian components associated with a particular state are approximately the same, the time efficient optimisation only requires the storage of a full covariance matrix at the state level, rather than the Gaussian component level[11]. This was the approach adopted in [2] (except a numerical optimisation scheme was used to find the model parameters).

5.2 Number of iterations required

For the semi-tied covariance matrix the estimation process is an iterative one. Figure 1 shows a typical change in auxiliary function value against iteration number, with the likelihood of a standard HMM (ie an identity semi-tied transform) shown for comparison. Here a time efficient scheme was run on a small system. The iteration number shown is the number of times all the model parameters are updated[12]. 10 full iterations were performed, however, almost all the gain is achieved after five iterations of updating all the model parameters. Between updates of all the model parameters the semi-tied transform, $\mathbf{A}^{(r)}$, is estimated using another iterative process. For this example, 100 iterations were used to estimate the semi-tied transform (with all other model parameters fixed), although the majority of the likelihood gain was obtained after around 10 iterations. These two distinct iteration stages are clearly visible in figure 1. For example, the likelihood gains between the zero and one marks indicate the gains as $\mathbf{A}^{(r)}$ is iteratively estimated with all other model parameters fixed. At the one mark the increase in likelihood as a result of updating the diagonal covariance matrices given the new estimate of $\mathbf{A}^{(r)}$ is shown. It can be seen that the majority of the gain in likelihood occurs during the first model update iteration where the log-likelihood increased from -139.1 to -136.8.

In many applications, particularly large-vocabulary speech-recognition tasks, the majority of the training time is spent obtaining the sufficient statistics from the training data, rather than estimating the model parameters given those statistics. Thus, the actual parameter estimation time is not crucial. However, the ability to guarantee stability and convergence are important, particularly for complex systems.

[11] Since the semi-tied transform is tied across many states, this is not the same as state-specific rotation.

[12] Of course strictly the mean is only estimated once.

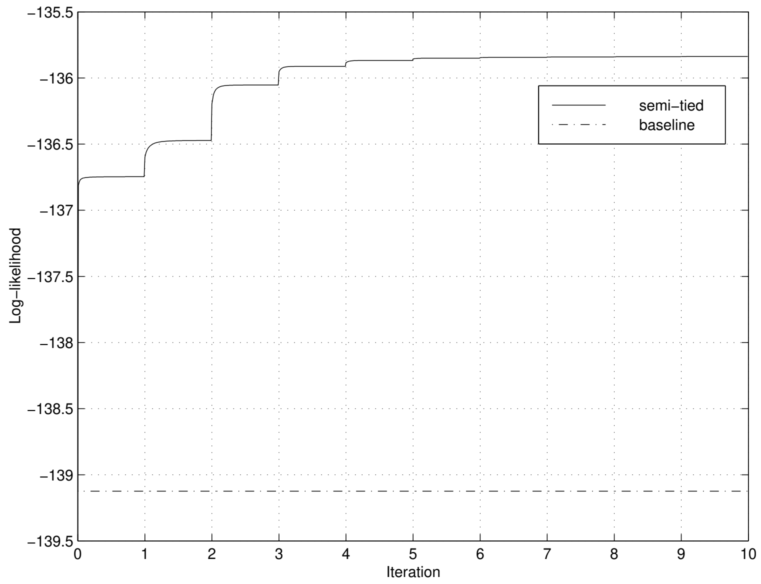

Figure 1: Increase in log-likelihood against iteration number when optimising a semi-tied covariance matrix is a time efficient mode. The likelihood obtained with a standard diagonal covariance HMM model set is shown as the baseline

5.3 Parameter tying

A variety of techniques have been used for clustering Gaussian components in speech recognition, for example decision tree tying [1]. Unfortunately, it is harder to decide how to group the components into semi-tied classes. The simplest approach is to tie all states together, or all the states of the same monophone together. The clustering may also be determined by generating a full covariance matrix single component system and performing agglomerative clustering.

An alternative scheme, that has previously been used for generating regression class trees, is based on locally maximising the likelihood [6]. Here a modified version of K-means clustering is used. If there are R semi-tied transforms then for each state[13], s, the semi-tied transform associated with that state, $\hat{r}^{(s)}$, is determined by

$$\hat{r}^{(s)} = \arg\max_{r\in R}\left\{ \sum_{m\in M^{(s)}} \gamma_m(\tau)\left(\log(|\mathbf{A}^{(r)}|^2) - \left(\mathbf{A}^{(r)}\mathbf{o}^{(m)}(\tau)\right)^T \mathbf{\Sigma}_{\text{diag}}^{(m)-1}\left(\mathbf{A}^{(r)}\mathbf{o}^{(m)}(\tau)\right)\right)\right\} \quad (32)$$

where[14] $\mathbf{o}^{(m)}(\tau) = \mathbf{o}(\tau) - \mu^{(m)}$. After the states have been reassigned, the semi-tied transforms may be re-estimated and the procedure repeated. This is guaranteed to generate a local maximum. One problem that has been observed with this form of optimisation is the dependency of the clustering on the start position, since there will be many local maxima.

For the experiments presented in this paper a single semi-tied transform is used for each monophone class and no reassignment of component or state to transform undertaken.

[13] This assumes that all components associated with a particular state will use the same semi-tied transformation.

[14] The component specific means and variances are no longer indicated with $\hat{\mu}$, as the values of these parameters are fixed for the assigning of states to semi-tied classes.

5.4 Numerical accuracy

When calculating the semi-tied transform there is a danger of the statistics stored not having full rank. This may be due to numerical inaccuracies or a limited amount of training data. Thus, when using equation 22, the inverse of $\mathbf{G}^{(ri)}$ may not exist. There are two solutions to this problem, similar to those used to ensure robustness in maximum likelihood linear regression (MLLR) [15]. The first is to use block diagonal transformations, thus dramatically reducing the chance of non-full rank matrices. Furthermore it decreases both the computational load (it is cheaper to invert three 13×13 matrices than one 39×39 matrix), and the memory requirements ($\mathbf{W}^{(m)}$ is now only required to be block-diagonal). Alternatively, singular value decomposition (SVD) may again be used for the inversion of $\mathbf{G}^{(ri)}$. In this work block diagonal transforms are used along with SVD to ensure robust inversions (although in practice SVD was not required).

5.5 Speaker and Environmental Adaptation

There is the question of how the model-based linear transformation schemes, such as MLLR [15, 10], which are currently popular in speech recognition, may be applied to model sets where semi-tied covariance matrices are used. MLLR may be applied to models with full covariance matrices [10]. This now involves solving

$$\mathrm{vec}(\mathbf{Z}) = \left(\sum_{m \in M^{(l)}} \mathrm{kron}(\mathbf{V}^{(m)}, \mathbf{D}^{(m)}) \right) \mathrm{vec}(\mathbf{W}) \tag{33}$$

where $M^{(l)}$ are set of Gaussian components associated with the linear transform to be estimated, vec(.) converts a matrix to a vector ordered in terms of the rows, kron(.) is the Kronecker product,

$$\mathbf{Z} = \sum_{m \in M^{(l)}, \tau} \gamma_m(\tau) \mathbf{\Sigma}^{(m)-1} \mathbf{o}(\tau) \xi^{(m)^T} \tag{34}$$

$$\mathbf{V}^{(m)} = \sum_{\tau} \gamma_m(\tau) \mathbf{\Sigma}^{(m)-1} \tag{35}$$

and

$$\mathbf{D}^{(m)} = \xi^{(m)} \xi^{(m)T} \tag{36}$$

$\xi^{(m)}$ is the extended mean vector such that

$$\hat{\mu}^{(m)} = \mathbf{W} \xi^{(m)} = \mathbf{A} \mu^{(m)} + \mathbf{b} \tag{37}$$

If implemented directly this is computationally expensive, both in terms of accumulating the statistics and generating the transforms. Techniques are available for reducing the computational load of accumulating the statistics [7]. An alternative scheme is to use a version of normalised domain MLLR, where the transform is generated in domain determined by $\mathbf{A}^{(r)}$. For further details of this type of adaptation and its limitations see [7].

These ML adaptation schemes may be contrasted with the least squares linear regression (LSLR) adaptation implemented in [12] when the decorrelating rotation described in [17] was used. Using LSLR it is not possible to guarantee that the likelihood of the adaptation data will increase. However the computational cost is far less.

6 Results

An initial investigation of the use of semi-tied covariance matrices was carried out on a large-vocabulary speaker-independent continuous-speech recognition task. All recognition experiments were performed on the 1994 ARPA Hub 1 data (the H1 task). The H1 task is an unlimited vocabulary task with approximately 15 sentences per speaker. The data was recorded in a clean[15] environment. No speaker adaptation was performed.

The baseline system used for the recognition task was a gender-independent cross-word-triphone mixture-Gaussian tied-state HMM system. This was the same as the "HMM-1" model set used in the HTK 1994 ARPA evaluation system [20]. In this model set all the speech models had a three emitting state, left-to-right topology. Two silence models were used. The first silence model, a short pause model, had a single emitting state which may be skipped. This model was used to represent short inter-word silences. The other silence model was a fully connected three emitting state model used to represent longer periods of silence. The speech was parameterised into 12 MFCCs, C_1 to C_{12}, along with normalised log-energy and the first and second differentials of these parameters. This yielded a 39-dimensional feature vector, to which cepstral mean normalisation was applied. The acoustic training data consisted of 36493 sentences from the SI-284 WSJ0 and WSJ1 sets, and the LIMSI 1993 WSJ lexicon and phone set were used. The standard HTK system was trained using decision-tree-based state clustering [22] to define 6399 speech states. For the H1 task a 65k word list and dictionary was used with the trigram language model described in [20]. All decoding used a dynamic-network decoder [18].

When generating the multiple component systems used for this task, *mixing-up*[16] was used [21]. The performance was investigated at various stages of this process. It should be emphasised that the grammar scale factor and insertion penalties were not optimised at any stage for the particular number of components in the system (or for the use of semi-tied covariance matrices). For the particular implementation of semi-tied covariance matrices considered here, all states of all context-dependent phones associated with the same monophone were assigned to the same semi-tied class. Also, both silence models were assigned to the same semi-tied class. Furthermore, a simple block-diagonal transformation was used. Thus the static, delta and delta-delta parameters had separate blocks associated with each of them. This resulted in very few additional parameters, 23322, in a system of up to 6 million parameters (the LIMSI phone set has 46 phones for English including silence).

The process of building the semi-tied covariance matrices was first to mix-up to the new number of components. Two iterations of Baum-Welch re-estimation were performed. The new semi-tied transform[17] was then estimated. Due to memory constraints only memory efficient estimation of the semi-tied transforms was performed (in terms of figure 1 this equates to a single iteration). Finally an additional two iterations of Baum-Welch re-estimation were run. This process was repeated as necessary. The first thing to notice about table 1 is that despite the very small increase in the number of parameters the effect on the recognition performance is quite dramatic[18]. For the single component case on the evaluation data, the use of semi-tied covariance matrices reduced the error rate by 12% with only a 5% increase in the number of parameters. For all cases the semi-tied covariance matrix case gave a performance gain over the standard covariance matrix; the performance of the standard 12-component system was achieved using only 6 components. In the 12-component case, a 12% reduction in word error rate was achieved, which is comparable with the performance achieved *with* incremental speaker adaptation for the standard system [10].

The performance figures in table 1 may be compared to the state-based rotation scheme [17].

[15]Here the term "clean" refers to the training and test conditions being from the same microphone type with a high signal-to-noise ratio.

[16]Mixing-up involves gradually increasing the number of Gaussian components in a particular state. The standard procedure is to take the Gaussian component with the largest weight, perturb the means to generate two components and retrain the system.

[17]For the implementation used for this work only the transform, or the mean and diagonal component specific parameters were estimated. When estimating the semi-tied transform the iterative schemes were run to convergence.

[18]The number of Gaussian components in table 1 indicates the number of Gaussian components in the speech states. The states modelling silence typically have more Gaussian components, for example in the 12 Gaussian component case all silence related states have 24 Gaussian components.

Number Gaussian Components	Semi-Tied Covariance	Distribution Parameters	Error Rate (%)	
			H1 Dev	H1 Eval
1	—	501018	13.93	15.54
	Block	(+23322)	12.27	13.70
2	—	1012938	12.01	13.04
	Block	(+23322)	11.06	11.81
4	—	2023980	10.56	11.43
	Block	(+23322)	9.86	9.65
6	—	3036918	10.08	10.91
	Block	(+23322)	9.17	9.30
8	—	4049224	9.67	9.97
	Block	(+23322)	8.88	8.61
10	—	5061530	9.42	9.51
	Block	(+23322)	8.46	8.38
12	—	6073836	9.57	9.20
	Block	(+23322)	8.62	8.12

Table 1: The performance of a standard system and a semi-tied covariance matrix system on the H1 development and evaluation data.

This scheme was implemented with separate transforms calculated for each state of each monophone using equation 2. The system was then trained by repeatedly mixing-up and Baum-Welch re-estimation in the same fashion as the standard system. On the H1 evaluation task this system had a word error rate of 14.25% for the single component system and 12.85% on the two component system. Although this shows a slight improvement over the standard system, the gain is considerably less than that achieved with the semi-tied covariance matrices described here. Furthermore the gains over the standard system became negligible as the number of components increased. It should be emphasised that the "state-specific" rotations estimated were far fewer than those in [17], which possibly explains the poor performance. However, there are still more transforms than in the semi-tied covariance matrix case.

The number of Baum-Welch re-estimations, after the semi-tied transform had been learnt, was set very low: only two. This is not expected to seriously effect the mixing-up process, but for a particular number of components it may give a worse recognition performance than is actually possible. Thus purely for recognition purposes an additional two iterations of Baum-Welch re-estimation were performed[19]. For the six component system this gave a slight increase in performance, 9.03% word error rate on the development data and 9.28% on the evaluation data.

An alternative method of using semi-tied covariance matrices is to take a fully trained system and then train the semi-tied transform. After obtaining the semi-tied transform additional iterations of Baum-Welch may then be applied. Using this approach on the 12 component system and performing two additional iterations of Baum-Welch gave 9.00% on the development task and 8.59% on the evaluation task. These again show improvements compared to the standard system, although not as large as incorporating training the semi-tied transforms into the mixing up process. This may partly be due to using only a single iteration to estimate the transform, in contrast to the mixing-up scheme where a new transform is estimated every time "mixing-up" is performed. However, this does show how the transforms may be incorporated when "mixing-up" is not used in the standard training procedure. Of course further improvements could be obtained by generating new components, typically using K-means clustering [12], given the current set of transforms, or by performing additional iterations to estimate the transform.

7 Conclusions

This paper has introduced a new form of covariance matrix, the semi-tied covariance matrix. Using this new form of matrix, it is possible to choose a compromise between the large number of parameters of the covariance matrix and the poor modelling ability of the diagonal case. Maximum likelihood re-estimation formulae are derived, which are guaranteed to increase the likelihood of the training data. The use of these re-estimation formulae for optimising heteroscedastic LDA transforms and full variance adaptation transforms is also discussed. Various implementation issues, such as memory and computational cost are detailed. How this new form of covariance matrix may be used with standard model-based adaptation schemes is briefly mentioned. The new models were tested on a large-vocabulary speech recognition task where a reduction in word error rate of 10% over the standard system was achieved with little increase in the number of parameters or computational cost.

Future work will involve experiments using the proposed linear adaptation scheme with semi-tied covariance matrices and use of other transformation groupings.

[19]The reason for only performing the additional iterations for recognition is to make the standard and semi-tied systems as comparable as possible.

Acknowledgements

Mark Gales was funded as a Research Fellow at Emmanuel College, Cambridge. The notation used for the full covariance MLLR transform was suggested by Olivier Cappé of ENST.

A Semi-Tied Transform Optimisation

This appendix considers the optimisation of the semi-tied transform given ML estimates of the mean and the current value of the diagonal element of the covariance matrix[20].

The objective is to maximise the following expression with respect to $\hat{\mathbf{A}}^{(r)}$ (from equation 20)

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}};\left\{\hat{\boldsymbol{\Sigma}}^{(r)}_{\text{diag}}\right\}) = \sum_{m\in M^{(r)},\tau}\gamma_m(\tau)\left\{\log\left(\left(\hat{\mathbf{a}}^{(r)}_i\mathbf{c}^T_i\right)^2\right) - \log\left(|\hat{\boldsymbol{\Sigma}}^{(m)}_{\text{diag}}|\right) - \sum_j\frac{\left(\hat{\mathbf{a}}^{(r)}_j\hat{\mathbf{o}}^{(m)}(\tau)\right)^2}{\hat{\sigma}^{(m)2}_{\text{diag}j}}\right\} \tag{38}$$

Rewriting equation 38 (all terms independent of $\hat{\mathbf{A}}^{(r)}$ are combined into K) yields

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}};\left\{\hat{\boldsymbol{\Sigma}}^{(r)}_{\text{diag}}\right\}) = \beta\log\left((\mathbf{c}_i\hat{\mathbf{a}}^{(r)T}_i)^2\right) - \sum_j\left(\hat{\mathbf{a}}^{(r)}_j\mathbf{G}^{(rj)}\hat{\mathbf{a}}^{(r)T}_j\right) + K \tag{39}$$

where $\hat{\mathbf{a}}^{(r)}_i$ is i^{th} row of $\hat{\mathbf{A}}^{(r)}$, the $1\times n$ row vector $\mathbf{c}_i$ is the i^{th} row vector of cofactors of $\hat{\mathbf{A}}^{(r)}$,

$$\beta = \sum_{m\in M^{(r)},\tau}\gamma_m(\tau) \tag{40}$$

and

$$\mathbf{G}^{(rj)} = \sum_{m\in M^{(r)}}\frac{1}{\hat{\sigma}^{(m)2}_{\text{diag}j}}\mathbf{W}^{(m)}\sum_\tau\gamma_m(\tau) \tag{41}$$

Differentiating with respect to $\hat{\mathbf{a}}^{(r)T}_i$ and equating to zero yields

$$\beta\mathbf{c}_i\mathbf{G}^{(ri)-1} = \mathbf{c}_i\hat{\mathbf{a}}^{(r)T}_i\hat{\mathbf{a}}^{(r)}_i \tag{42}$$

It is simple to see that $\hat{\mathbf{a}}^{(r)}_i$ must be in the direction of $\mathbf{c}_i\mathbf{G}^{(ri)-1}$. Letting $\hat{\mathbf{a}}^{(r)}_i = \alpha\mathbf{c}_i\mathbf{G}^{(ri)-1}$ gives

$$\beta\mathbf{c}_i\mathbf{G}^{(ri)-1} = \alpha^2\mathbf{c}_i\mathbf{G}^{(ri)-1}\mathbf{c}^T_i\mathbf{c}_i\mathbf{G}^{(ri)-1} \tag{43}$$

This is now a simple linear optimisation problem in α. Hence

$$\alpha = \pm\sqrt{\left(\frac{\beta}{\mathbf{c}_i\mathbf{G}^{(ri)-1}\mathbf{c}^T_i}\right)} \tag{44}$$

Only the positive root is considered[21], hence the final solution for row i is

$$\hat{\mathbf{a}}^{(r)}_i = \mathbf{c}_i\mathbf{G}^{(ri)-1}\sqrt{\left(\frac{\beta}{\mathbf{c}_i\mathbf{G}^{(ri)-1}\mathbf{c}^T_i}\right)} \tag{45}$$

The optimisation is thus an iterative one, where each row of $\mathbf{A}^{(r)}$ is optimised given the current value of all the other rows.

Thus by using the current estimate of the cofactors, each row of $\mathbf{A}$ may be optimised independently. By then iteratively running through the rows the complete transform may be optimised efficiently and robustly. Each iteration is guaranteed to increase the likelihood of the training data.

[20] An alternative optimisation scheme was given in the original presentation of semi-tied covariance matrices [9]. In all cases the two schemes converged to the same solution, however the scheme presented here is felt to be more elegant.

[21] It makes no difference whether the positive or negative root is selected as they will yield the same likelihood.

References

[1] L R Bahl, P V de Souza, P S Gopalkrishnan, D Nahamoo, and M A Picheny. Context dependent modelling of phones in continuous speech using decision trees. In *Proceedings DARPA Speech and Natural Language Processing Workshop*, pages 264–270, 1991.

[2] S Chen, M J F Gales, P S Gopalakrishnan, R a Gopinath, D Kavensky, P Olsen, and L Polymenakos. IBM's LVCSR system for transcription of broadcast news used in the 1997 hub4 english evaluation. In *Proceedings ICASSP*, 1998.

[3] S B Davis and P Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions ASSP*, 28:357–366, 1980.

[4] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[5] K Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1972.

[6] M J F Gales. The generation and use of regression class trees for MLLR adaptation. Technical Report CUED/F-INFENG/TR263, Cambridge University, 1996. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[7] M J F Gales. Adapting semi-tied full-covariance HMMs. Technical Report CUED/F-INFENG/TR298, Cambridge University, 1997. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[8] M J F Gales. Maximum likelihood linear transformations for HMM-based speech recognition. Technical Report CUED/F-INFENG/TR291, Cambridge University, 1997. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[9] M J F Gales. Semi-tied full-covariance matrices for hidden Markov models. Technical Report CUED/F-INFENG/TR287, Cambridge University, 1997. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[10] M J F Gales and P C Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, 10:249–264, 1996.

[11] A J Hewett. *Training and Speaker Adaptation in Template-Based Speech Recognition*. PhD thesis, Cambridge University, 1989.

[12] D M Hindle, A Ljolje, and M D Riley. Recent improvements in the AT&T speech-to-text (STT) system. In *Proceedings ARPA Speech Recognition Workshop*, 1996.

[13] B H Juang. Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, 64:1235–1249, 1985.

[14] N Kumar. *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*. PhD thesis, John Hopkins University, 1997.

[15] C J Leggetter and P C Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9:171–186, 1995.

[16] L A Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Transactions Information Theory*, 28:729–734, 1982.

[17] A Ljolje. The importance of cepstral parameter correlations in speech recognition. *Computer Speech and Language*, 8:223–232, 1994.

[18] J J Odell, V Valtchev, P C Woodland, and S J Young. A one pass decoder design for large vocabulary recognition. In *Proceedings ARPA Workshop on Human Language Technology*, pages 405–410, 1994.

[19] L R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, February 1989.

[20] P C Woodland, J J Odell, V Valtchev, and S J Young. The development of the 1994 HTK large vocabulary speech recognition system. In *Proceedings ARPA Workshop on Spoken Language Systems Technology*, pages 104–109, 1995.

[21] S J Young, J Jansen, J Odell, D Ollason, and P Woodland. *The HTK Book (for HTK Version 2.0)*. Cambridge University, 1996.

[22] S J Young, J J Odell, and P C Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings ARPA Workshop on Human Language Technology*, pages 307–312, 1994.